CANMET

Canada Centre for Mineral and Energy Technology

Centre canadien de la technologie des minéraux et de l'énergie

Energy Research Laboratories

Laboratoires de recherche sur l'énergie

Energy, Mines and Resources Canada

Énergie, Mines et Ressources Canada

Canada

THE ENERGY OF OUR RESOURCES • THE POWER OF OUR IDEAS

L'ENERGIE DE NOS RESSOURCES • NOTRE FORCE CREATRICE

91-07105/

Development of Process Control Software; Phase II

Final Report by

Atlantis Scientific Systems Group Inc.

Work on the project was conducted under the auspices of
the Canadian Centre for Mineral and Energy Technology,
Energy, Mines and Resources Canada.
This work was supported by the Federal Panel on Energy R&D (PERD).

DSS FILE NO.: 003SQ.23440-9-9111
DSS CONTRACT SERIAL NO.: 23440-9-9111/01-SQ
CANMET PROJECT NO.:
CANMET SCIENTIFIC AUTHORITY: T. J. W. de Bruijn
DATE: May 31, 1991

Executive Summary of the Development of Process Control Software; Phase II. Contract No. 23440 9 9111

Atlantis Scientific Systems Group Inc.
1827 Woodward Drive, Ottawa, Ontario, K2C 0P9

May, 1991

This contract involved the further development of a process monitoring and control software package, developed by Atlantis Scientific. The developments comprised of the following. A remote communications capabiltiy that allows the system to be monitored and controlled from a remote location. An enhanced schematic editor to allow cut and paste. A cascadable control loops, to provide more sophisticated heater control. The software changes were made in several stages, with each stage concluded by an on-site test. Finally, a manual for the new features of PC Control was produced.

Sommaire du travail accompli pour le contrat numéro 23440 9 9111

Atlantis Scientific Systems Group Inc.
1827 Woodward Drive, Ottawa, Ontario, K2C 0P9

mai, 1989

Ce contrat porte sur l'expansion du logiciel de commande de processus, logiciel qui a été conçu par Atlantis Scientific. Les tâches accomplies sont les suivantes. La capacité de communiquer à longue distance pour permetrre au système de pouvoir être surveillé et controllé d'un site autre que le site où se situe le système a été ajoutée. L'éditeur a été amelioré pour permettre à l'usager de choisir un bloc de texte et de le changer de place ou le copier. Des circuits de controle en série ont été ajoutés pour permettre un controle plus précis de la température. Le logiciel a été modifié par étapes et après chaque changement, le logiciel a été soumis à des essais. Un nouveau manuel pour le logiciel de contrôle a été redigé.

Final Report of the Development of Process Control Software, Phase II, Contract No. 23440 9 9111

Atlantis Scientific Systems Group Inc.
1827 Woodward Drive, Ottawa, Ontario, K2C 0P9

31 May 1991

Contents

1 Introduction 1

2 Task 1: Communication and Control of the Plant from a Remote Location 1

3 Task 2: Improve Screen Editor, Layout, Schematics 2

4 Task 3: Off-line Graphing Facility 2

5 Task 4: Cascadable PID Control Loops 2

6 Task 5: Training and Test Support 4

7 Task 6: Revision of the User Manual 4

A Extract from Original Investigative Report: 5

A.1 Introduction 5

A.2 Off-the-Shelf Software and Evaluation of Communication Libraries for PASCAL . . . 5

A.3 Issues Relating to Incorporation of Remote Communications into PC Control 6

A.4 Conclusions 9

1 Introduction

This document describes software design and development related to the above contract, and provides information on where exact contractual obligations were modified or exceeded, only for the purposes of enhancing the software. The use of the features now in place in PC Control are described in the article *Addendum to PC Control Manual*, delivered to the CANMET Scientific Authority in May, 1991. The initial contract contained several tasks, which were subsequently changed or moved over to a future contract. For details, examine the contents of the contract amendments.

All work described in the amended contract was completed by May 31, 1991, on schedule. The implementation of each of the required tasks are described in the following sections.

2 Task 1: Communication and Control of the Plant from a Remote Location

This task was subdivided into an investigative and an implementation task. After a thorough investigation, a report entitled *Monitoring and Controlling the Hydrocracking Unit From a Remote Location - An Investigation* was presented to the Scientific Authority. This report is included at the back for reference.

The investigation showed that this feature is feasible, and the report outlined the design of the remote communications software. In addition, a portable computer and high speed modems were specified from numerous models on the market, to meet with CANMET's special requirements.

After reviewing the investigation report, the Scientific Authority gave a confirmation to continue with the implementation phase. Following this, a portable computer and high speed modems were acquired, along with utility and communications library packages needed for the software development.

Since the next task involved using a higher density display mode for the graphics card, the current task would also be affected. Hence a part of the next task was accomplished earlier than expected, and the resulting intermediate version of the software was delivered to CANMET.

While this early enhanced PC Control program was being used and tested, the remote communications development began, and continued in many stages. The programs were developed outside of the PC Control software to ease design and testing. The first stage was to simply transmit screen contents from one computer to another without the colour attributes. Once this was tested, the colour attributes were added too. Next, the software was modified to allow keyboard characters typed at one computer to be transmitted and received by another computer, while the screen contents were being transmitted by the host computer. Finally, the file transfer part of the program was written. Each small function in the remote communications program was written and tested before continuing with the next.

In general, the plan remained as described in the investigative report, except for some small improvements. One such change was to develop a sophisticated algorithm to detect and transmit only the changed characters on the screen, thus speeding up the communications. Another change was to enhance the XMODEM protocol for file transfer, which can now transmit file names and

exact file sizes.

The remote communications package was added to the PC Control program only after having been tested as a "stand-alone" program with various modem speeds and IBM compatible computers and video graphics cards.

The remote communications version of PC Control was then delivered to CANMET and tested. Since the time of writing this report, the program has been functioning correctly for over 3 months. The resulting PC Control program greatly extends the usability of the plant by letting users monitor and control it after work hours.

3 Task 2: Improve Screen Editor, Layout, Schematics

There were three goals to be achieved in this task. First, to allow 43-line or 50-line by 80 character display for computers having an EGA or VGA graphics board respectively. This task was implemented early in the contract. Since the addition of more equipment to the plant resulted in a more complicated schematic diagram, this task needed to be implemented first.

The second and third objectives of the screen editor were implemented together. As with many commercial editors, these two features achieved the equivalent of cutting and pasting of sections of a schematic diagram. The intermediate step between cutting and pasting is to save or recall the schematic from a disk file. Hence, a library of parts can be built and reused. The screen editor was designed to function with any number of 80 character lines, including the 43 and 50 lines offered by EGA and VGA graphics boards respectively.

4 Task 3: Off-line Graphing Facility

This task was removed from the original contract, to another future contract.

5 Task 4: Cascadable PID Control Loops

The original object of this task was to achieve unattended control over the internal temperatures in the reactor vessel. Once implemented, it would provide another level of automation, freeing plant personnel of this duty.

As per the contract amendment, a more generalised system was required that would also help in future control mechanisms that involve calculation of set points from measured inputs.

To answer both these requirements, our implementation involved the addition of a feature that calculates set points of a control loop from the values of an output or a measured input point. If the set point of one control loop is calculated from an output point that is part of another control loop then this represents cascaded control in the normal sense of the word. For this reason, this feature is known as cascading.

In designing this feature into the PC Control software, the main objective was to remain compatible with the old tag structure. When the design was finished, the same tag structure remained,

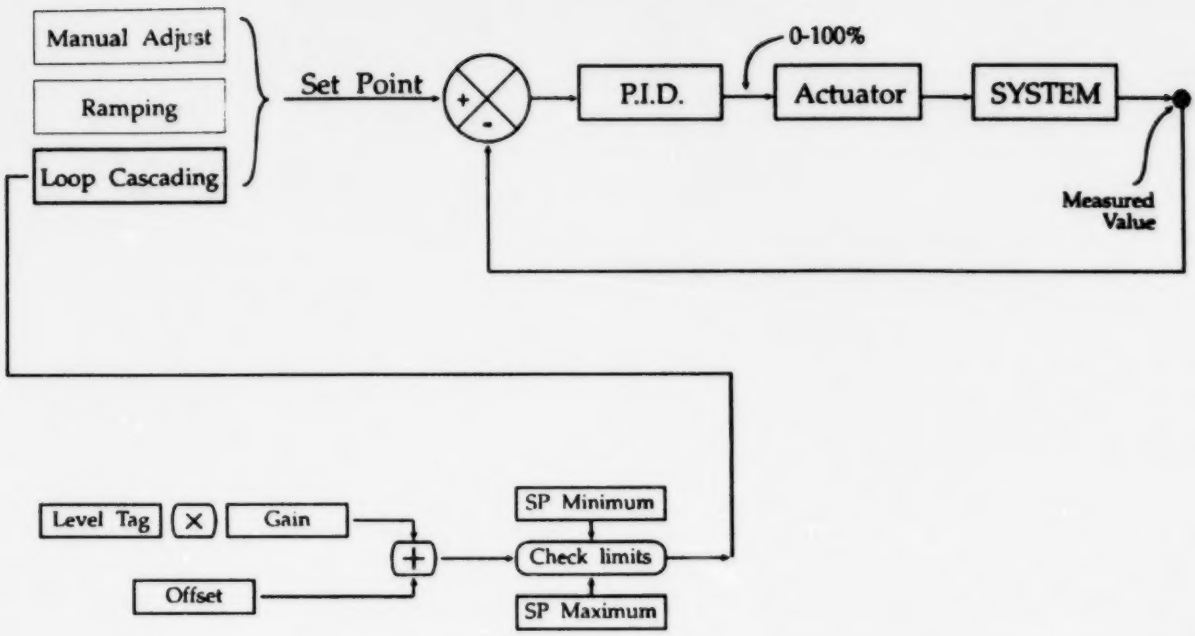

Figure 1: PID Control Loop with Cascaded Set Point

except that the total number of loops for one control output tag was reduced from 3 to 2. This was done to accommodate the various cascading parameters within the same memory space that the old tag structure used. In all cases to date, only one loop has ever been used, and so the reduction to 2 will have no impact for current users of PC Control.

The cascading parameters are stored in that level output tag whose set point needs to be automatically calculated. The cascading parameters are: Gain, Offset, Set Point Minimum, Set Point Maximum and Cascade Tag, described in the following.

To provide flexibility, both a gain and offset parameters are used to scale the value of the tag to be cascaded, to yield the new set point. The mathematical expression is as follows:

$$\text{Set Point} = (\text{Tag Value} \times \text{Gain}) + \text{Offset}$$

The resulting set point is compared against the user specified minimum and a maximum set point limits. If the set point falls between these limits then the new set point is accepted, otherwise the old set point remains in effect. The operation of the cascading feature is shown in Fig. 1.

When a control loop is active, the tag structure member **.algno** is given the value of 3 (this being the historic value). And when the loop is not active **.algno** has the value 0, as always. When cascading is enabled, the (new) value is 4. The new code which implements cascading is adapted from the old code, and is made more compact and time efficient. The cascading function is a superset of normal PID control, in that the function first calculates a new set point, and then calls the existing PID control algorithm.

Once coded, the new program was well tested in a demo program, before being used in the plant. The program was first tested without the cascading feature enabled, and was found to work properly.

Next the four heater controls to the reactor were set up to use cascading. During some initial tests, good control was obtained on the internal reactor temperatures. However, the system was still being fine tuned as this report was written.

6 Task 5: Training and Test Support

Under this phase, training and test support for PC Control was provided as and when required. The final version of the software was installed and tested under this phase.

7 Task 6: Revision of the User Manual

The latest version of PC Control has the version number 3.60. A companion manual called *Addendum to PC Control Manual* was written to accompany the original manual, *PC Control User's Manual*. It provides users instructions on using the new features, and describes various changes and fixes to PC Control.

A Extract from Original Investigative Report:

Monitoring and Controlling the Hydrocracking Unit From a Remote Location – An Investigation

Atlantis Scientific Systems Group Inc.
1827 Woodward Drive, Ottawa, Ontario, K2C 0P9
December, 1989

A.1 Introduction

This article reports the findings of an investigation to incorporate remote monitoring and control capabilities to the current software, PC Control. Both the hardware and software requirements are assessed. The function of monitoring and controlling the hydrocracking unit from a remote location via modems and telephone lines will add greatly to the use of PC Control, especially during long runs when a plant operator is not always present on-site. Even when a plant is being supervised, it would be helpful to be able to monitor the process from another building. Additionally, data files generated by the monitoring and control program will be available for 'downloading' to a remote computer for further data analysis at the remote computer. Access to the plant (or host) computer via a remote computer requires four main functions:

- Transmission of screen characters to the remote computer
- Reception of typed characters from the remote computer
- Transmission of data files to the remote computer
- A password protection scheme to prevent unauthorised remote access

The PC Control software which runs the hydrocracking bench scale unit consists of two parts: a background program to monitor and control the process, and a foreground program to set up monitoring/control parameters. When the remote access capability is implemented, the laboratory computer will run an enhanced version of the current foreground program, which in addition to its normal tasks, will also communicate with a connected modem. This will leave the important background program undisturbed.

The remote computer will run a special program to display text screens just as seen on the host computer, and to send keyboard commands to the host computer.

When the remote access capability is added to PC Control, the possibility of unauthorised access to the plant becomes an important issue, and must be addressed by means of password protection.

A.2 Off-the-Shelf Software and Evaluation of Communication Libraries for PASCAL

There exists on the market today, software products such as REMOTE-2, which allow remote access to a host computer via modems. The REMOTE-2 software works by making a communications program

'memory resident, and interrupt-driven' in the host computer. For our purposes, this is not a viable solution to the remote access requirement because the background program for monitoring and control is itself memory resident and interrupt driven. The background program relies on the timer tick interrupt (number 1C hexadecimal) to be periodically 'serviced' by the computer. While the two programs might make use of different 'interrupt channels', it is difficult to predict what the interaction of two memory resident, interrupt-driven programs is for the synchronous data acquisition and control task of PC Control's background program. Furthermore, REMOTE-2 software will have a bigger memory requirement since it has more communications functions in its program than are required for our purposes. This is an important consideration because the DOS operating system machines put a maximum limitation of 640 kilobytes of useable memory space. It is for these reasons that we do not intend to use available software such as REMOTE-2.

Our approach will be to add program code needed to accomplish the four basic functions outlined in the previous section to the foreground, user-interface of PC Control. Since the user-interface is already active (loaded into memory) at the time of remote access, only the least amount of program code for communications will be added. An added advantage will be that the remote user will be familiar with the menu system of PC Control when accessing it from a remote terminal, and will not have to learn a new piece of software. Other advantages will be apparent in the later description of the task.

The communications program code needs to be flexible, yet fast, and rather than write the complex subroutines required for our purposes, we will opt for off-the-shelf library routines, which are efficient since they are written mostly in assembly language. A common communications library package for Turbo PASCAL is 'ASYNCH PLUS' from Blaise Computing Inc. This package has been acquired for examination, and appears to be adequate. Included in the library are the necessary functions to achieve the popular 'XMODEM' and 'XMODEM with CRC' method of file transfer from one computer to another.

A.3 Issues Relating to Incorporation of Remote Communications into PC Control

Introduction

The changes and additions to the existing code necessary to accomplish remote communications are detailed in 'pseudo code' included in the appendix section of the original report. There are two parts: REMOTE and HOST, the first being the stand alone program which will run in the remote computer. The second is a series of new subroutines and changes to be added to PC Control's foreground program. The program code that makes up REMOTE and HOST is interdependent, and we need only describe the methodology in one part, HOST.

At the remote location, the user chooses a command in the REMOTE program to 'dial'. The REMOTE program sends instructions to the remote modem to dial the laboratory phone number. If there is a modem in the laboratory (host modem), then both modems send a tone of a specific frequency to identify that each is a modem. A normal telephone may be connected to a modem whenever remote operation is not in progress, but this telephone is switched out automatically whenever a modem goes into operation.

If the two modems can synchronize with each other in terms of the baud rate (communication speed

in bits per second), then this state is known as data carrier detected, and is known to each computer. Next, the host computer will prompt the remote computer for user name, and then user password. Each of these can be strings of characters that will be established beforehand. When both strings match, then the foreground program can proceed. The foreground program will set a flag (program memo) to state that remote communications is active, and will remain so until the carrier is lost. The carrier is lost when the remote user hangs up, or there is a modem or telephone line problem.

The ongoing part of the foreground will then begin to service two new functions. The first is to send a copy of the current screen contents over to the remote computer, and the second is to receive keystrokes typed at the remote location. With these two functions, the remote terminal will have the same display and command capabilities as the host computer. This includes moving from menu to menu with keyboard entries that normally would originate from the host computer. The remote user will be able to use the different editor, create new macro files, examine time queues, and so on. Almost every function will be accessible to the remote user.

The exceptions will be as follows. First, the user will not be able to use the Quit menu item of the foreground program, since doing so would mean terminating not only the remote communications, but also the other critical foreground functions. These functions include the periodic data logging and printing activities to disk and printer respectively. Secondly, both the real time XY/Bar graphics and the trend graphics capabilities will be inaccessible. When in graphics mode, the video screen is manipulated by turning on and off pixels of which there are too many to be transmitted in a reasonable time. However, a future solution to this may be to send data values of the 6 parameters to be graphed at the remote location. But for now, the two function keys <F3>, and <F5> will not result in graphics either at the host or the remote computer, whenever the remote communications is active.

Transmission of Text Screen

The transmission of the text screen contents from the host to the remote will be one of the more difficult tasks. Because of an effective transmission rate of just 240 bytes per second, a screen of 80 columns by 25 rows of colour characters would take about 17 seconds for transfer via the serial line. This means that the data values will not update at the remote terminal quicker than 17 seconds. However, since most of the screen is the schematic, which remains constant, upon which data values are superimposed, we need only send these changes once the full screen has been sent.

This can be achieved by efficient algorithms, resulting in dramatic decreases in the update time. We calculate that the worst case where all data values have changed (which we estimate to be 50, with 4 characters per data value) will need 1.7 seconds for transmission. This time interval is acceptable, considering that the data values update every 2 seconds anyway, but because these two processes cannot be coordinated, data values will probably update every other sample. After incorporating some error checking into the transmission scheme, the update rate may drop to half of the rate calculated above. This implies that a modem of 4800 baud or greater is desireable, providing that the telephone lines allow this speed (See Section 4). The screen transmission method described above assumes that both the colour and the character code are sent for every character. Other schemes have been thought about, which can further reduce transmission time, and allow transmission error checking.

The writing of characters to screen by the foreground program in the host computer occurs via direct writes to the screen memory or by using certain PASCAL library calls. In both cases, the end result is

that the screen memory is written to. This is where the remote communications routines should look for screen contents, and be able to automatically adjust for the different video boards. Another task of the contract to upgrade the software requires a higher density mode for text display. In the case of the EGA and VGA video adaptor boards, their text mode can be chosen to have 25 rows of 80 columns or a higher number of rows than 25. The EGA video adaptor can support 43, while the VGA adaptor can support 50. The remote communications functions must be able to detect video modes, video adaptor board types, video memory addresses and so on, and support those varying parameters accordingly.

Command Via Remote Keyboard

The next important task is to provide command capabilities from the remote terminal. This involves letting the keyboard of the remote computer substitute for the keyboard of the host computer. When this task is being tested, we may decide to allow both keyboards to input, so that a user at the host computer can step in to correct problems that might arise from communications problems.

The foreground program of PC Control uses different keyboard subroutines, from several parts of the program. One of the first tasks will be to correct this situation so that calls to the different keyboard subroutines instead make use of just one particular subroutine, called GetC. Once this is done, then the GetC subroutine can be appended so as to receive keyboard characters not only from the host keyboard, but also received characters that the remote keyboard had typed.

When the foreground program prompts a user for some input, it enters into a state where other functions are frozen for a while. For example, when entering an action command after pressing <F2>, a prompt appears at the bottom of the screen, and the computer waits to accept a string of characters until a carriage return is found. Meanwhile, the menu system is inactive, the tag data values do not update and a remote user cannot log onto the host system. Now, if it is the remote computer communicating with the host, and attempting to type in an action command or other data request, then it must be completed in full, otherwise the system keeps on waiting. This can be a problem if the carrier is lost due to modem or telephone line problems. To prevent this, we must be able to detect the loss of a carrier and if that happens then the system should return to a 'normal state' so that a remote user can dial in again.

File Transfer

Finally, we have to incorporate a means of file transfer from the host computer to the remote computer. This process is known as downloading. We can do this neatly if we add an additional item under the Data menu of PC Control: 'Download'. When the remote user picks this item, he is queried for the path name (directory). Then the user will specify the extension name of the file. For example, the user will type in 'PRN' for the Lotus compatible data files. After a valid path and extension names have been given, all matching file names with the given specification will be presented in a menu. The user can pick the one he wishes to download. When this file is downloaded, it will have the same file name as the host, but in the default directory of the host computer. Many methods of transferring files exist, but a recognised reliable means is by the XMODEM protocol. The ASYNCH PLUS software library provides this and a more robust, cyclic redundancy check (CRC) version of subroutines to accomplish this task.

File transfer is time consuming - we will attempt to provide time estimates before file transfers begin.

However, the user must note that other foreground activities such as data logging and reporting will be prevented during downloading. For this reason, a suitable time must be picked for file transfer, so as not to disturb pending logging and reporting activities.

A.4 Conclusions

It is our opinion that the current data acquisition and control software, PC Control can be modified and then enhanced as described above to allow a remote computer to be linked with the plant computer via modems and a telephone line. The remote computer will be able to display the text oriented screens as visible on the plant computer's video monitor. In addition, commands and process changes will be allowed to be typed at the remote location, which will then be executed by the plant computer. Finally, we will also be able to add a facility to 'download' disk files from the plant to the remote computer.

For use as a remote computer, we recommend the Toshiba T5200 portable computer with a 100 Mbyte hard disk. The modems required are Hayes V-series 9600 baud Smartmodem. An initial list of additional software and hardware requirements are shown in Table 7 and Table 8.